The Main Menu

Newsletter for AppleWorks Users

Volume 2, Number 3 March 1987

How Much Time Do You Waste?

Time and Date Math in AppleWorks

by Steve High

Okay, class, it's word-problem time: If your boss began lunch at 11:47 a.m. and got back to work at 2:18 p.m., how much time did he waste? And is this a good time to ask for raise?

AppleWorks can't help you with your raise, but it can compute the time elapsed between two events—if you teach it how. If you're paid by the hour, it's important for you to know the number of hours you spent on a certain task; and the same principles used to subtract starting and ending times can be used to determine the number of days elasped since you sent a particular bill.

►Dates to Numbers

The basic problem is converting dates and times to numbers and back again. AppleWorks date format is MAR 15 87—nine characters, including spaces. Time takes the form 12:00 PM—eight characters. Note the letters MAR and PM: Because they're not numbers, AppleWorks can't calculate them. The solution is to convert them to numerals and put them into your spreadsheet. This involves several file transfers between the database and spreadsheet and, interestingly enough, between both modules and the word processor. A macro program and a RAM disk are almost required here because of repetitive pathname typing.

Continued on p. 6

Inside

Heard Here First

by Deborah de Peyster

The 2.0 standard: There seems to be an informal movement among manufacturers of AppleWorks-enhancement products to rally their support around version 2.0. According to one developer, "We've all set our standards on 2.0. It's an unstated protocol that to get along we'll all work in 2.0." **Applied Engineering**, **Checkmate**, and **Beagle Bros** are a few of the developers mentioned as supporting 2.0 as the new standard. That means that all their future products will work together, but it also means that you'll have to upgrade to version 2.0 to take advantage of them. They won't work on older versions of AppleWorks.

If you've got everything working with AppleWorks just the way you like it now, don't worry about changing. But if you want to take advantage of these new products, you should think about switching before the end of April, when Apple's $50 upgrade offer ends.

Speaking of interesting new products: We understand Beagle Bros, of San Diego, California, is working on a program that merges the word-processing power of AppleWorks with the graphics ability of The Print Shop.

The new package, scheduled for release in early summer, is called **Variety Page**. With its added commands, it turns AppleWorks into a mini-desktop-publishing program. It works by letting you call Print Shop clip art into text when printing. Although the graphics images won't show on screen, they'll appear in the printed document where specified. Variety Page also gives you the ability to put your word-processing text into two columns and print it in that format.

We think it sounds like an interesting program, especially since it combines the best of two top applications—**AppleWorks** and **The Print Shop** —in one bundle. We'll pass along the details as we learn more.

Macro power: AppleWorks power users trade hints for building efficient macros like investors who banter for one-upsmanship on the latest hot stock. But now even those of you who may have purchased your first macro product only yesterday can share tips with the best builders around. And that's thanks to *The Main Menu*'s own Steve High, who, in collaboration with Pinpoint Publishing, will introduce **Playmaker**, a disk containing numerous macros already designed and ready to go. Playmaker works with Pinpoint's macro program KeyPlayer and should be available by late spring for about $20–$30.

Make living with pathnames bearable: The author of MacroWorks, Randy Brandt, has developed a little program called **Pathfinder** that adds some new menu selections to AppleWorks to help you avoid pathnames when creating subdirectories.

Load Pathfinder and go to AppleWorks' Other Activities menu. Item 5, which used to say, "Format a blank disk," now says, "Pathfinder." When you select it, it gives you three menu choices: Add a subdirectory, drop a subdirectory, and rename files.

Before Pathfinder came along, you had to remember or find the pathname of the file you wanted, then type it carefully to create a subdirectory. Pathfinder lists all your files; all you do is select them to create subdirectories. The program deals with the pathnames—you don't have to. It's a great tool for people who use hard disks or UniDisks, Brandt says, because it speeds up operation noticeably.

Brandt will sell PathFinder on his own, not through Beagle Bros. It's priced at $20, including shipping and sales tax; it works with AppleWorks 2.0 and all 2.0 accessory programs. You can reach Brandt at JEM Software, 745 East Bradley #31, El Cajon, CA 92021.■

Sorry!

The February 1987 feature "Three-Label Printing in AppleWorks" contains four errors. The corrected steps should read as follows:

10) Press OA-D, type 6, and press Return.
13) Type 4 and press Return.
14) Type 2 and press Return.
22) Press OA-P, type 3, and press Return.

Answers

by Kelly Stirn

If you'd like Kelly to answer your AppleWorks questions, please write him at The Main Menu, *Elm Street, Peterborough, NH 03458. We'll print the answers to as many questions as we can.*

The Main Menu ►►►►►►►►►►►►►►►►►►►►

Deborah de Peyster, *Editor*
Steve High, *Contributing Editor*
Kelly Stirn, *Contributing Editor*
Paul Statt, *Contributing Editor*
Eileen Terrill, *Managing Editor*

Kelly Stirn is the AppleWorks engineering product manager for Apple Computer Inc. Steve High was a member of the AppleWorks development team and now owns Steve High Public Relations. One of his clients is Applied Engineering Inc.

The Main Menu (ISSN 0890-2585) is published monthly by CW Communications/Peterborough, Inc. 80 Elm Street, Peterborough, NH 03458. U.S. subscription rates $29.97, one year. Application to mail at second class postage rates is pending at Peterborough, NH, and additional mailing offices. Entire contents copyright 1986 by CW Communications/Peterborough, Inc. No part of this publication may be printed or otherwise reproduced without written permission from the publisher. POSTMASTER: send address changes to The Main Menu, 80 Elm Street, Peterborough, NH 03458. (800) 441-4403.

Disk Space vs. RAM

Q How can a document on the desktop be larger than it is on disk? The discrepancy increases the more memory I have in the Apple.

A *AppleWorks organizes data in memory—and the AppleWorks desktop is plain random-access memory (RAM)—differently from the way ProDOS does on disk. AppleWorks divides RAM on a 1-megabyte card, for example, into 52,000 pieces. Each piece contains 20 bytes of data—for a total of 1,040,000 bytes, or 1 megabyte. The first 4 bytes of each piece tell where the next 20-byte chunk is, while the rest are AppleWorks data. The more memory, the more pointers AppleWorks needs to access it.*

To make AppleWorks as fast as it is, though, Bob Lissner had to make each piece actually 32 bytes long. AppleWorks decides which data to use next by multiplying the number of bytes in the first piece by the number in the second piece—it's not what you or I would do, but it works for AppleWorks. AppleWorks is also unlike us in that multiplying by 32 is easier for it than multiplying by 20, because of the hexadecimal system. The 32-byte chunks don't always divide bytes efficiently.

The problem is it takes many more instructions in assembly code to multiply the number 20 than it does to multiply the number 32. More instructions in assembly code mean more time. This is especially apparent, for example, when reformatting a document or deleting one from memory.

So, what does all this mean for the user? When you load a file into memory—that is, onto the desktop—AppleWorks places your data into 32-byte pieces. Some of these pieces may be wasted. For example, if a word-processor line is 65 bytes long, it will take up 96 bytes of memory; 65 bytes equals 2 × 32 bytes plus 1. To get that 1 extra byte, the AppleWorks memory manager grabs another 32-byte piece of memory. Although this particular line may take up 65 bytes of disk space, it will take up 96 bytes of RAM.

The likelihood of a file taking up more memory is greater with larger memory-expansion cards, since more bytes are required to point to further points in memory. Basically, just remember AppleWorks stores data in 32-byte chunks on the desktop—that is, in RAM—and in 20-byte chunks on disk.

ImageWriter Superscripts

Q When printing a document with superscripts on my ImageWriter printer, these characters sometimes aren't positioned above the line of text. Why is AppleWorks inconsistent?

A *That's a problem with your ImageWriter, not AppleWorks. To get superscripts and subscripts on the ImageWriter, AppleWorks sends a sequence of codes that tells the printer to reverse-feed the paper a fraction of a line, print the character, then forward-feed the paper back to its original position. The problem is slack in the paper and sometimes in the ImageWriter gears that roll the platen. Slack sometimes makes the paper stay in the same position, even though the gears have attempted to move the platen a fraction of a line.*

The ImageWriter creates superscript and subscript characters this way because commands for true superscripts and subscripts aren't available. The good news is that if you have an ImageWriter II, these commands are available. If you use these printer options often enough, you may want to create custom print for your ImageWriter II with the following commands:

superscript begin	*= Escape x*
superscript end	*= Escape z*
subscript begin	*= Escape y*
subscript end	*= Escape z*

Certain programs, such as Beagle Bros PowerPrint, download special superscript and subscript character sets to your ImageWriter to be used by AppleWorks. Anyone know of any others?

Misaligned Columns

Q I've created a word-processor document by copying columns of data from the spreadsheet. I like to use my printer's proportional character set, but when I print this document, the columns no longer line up. Can I fix this?

A *Sorry, there's no way to solve this problem. The rule is never print columnar data with proportional characters, or in justified mode. Let me explain why. For those models whose proportional printing AppleWorks supports directly, each character takes up a different amount of space on paper. The character* i*, for instance, takes up a lot less space than the character* m*. The particular characters you use to construct the first column of data determine the beginning of the next column.*

AppleWorks can't show proportional characters on screen,

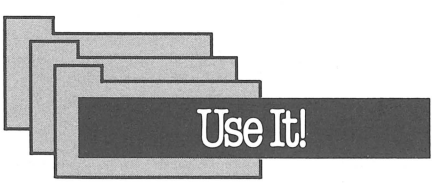
since every character takes up the same amount of screen space. AppleWorks does show you accurately the end of each line.

You'll run into the same problem of misaligned columns if you attempt to justify a column of text. When AppleWorks justifies a line of text, it places incremental spaces between words to stretch the line to the right margin. AppleWorks uses a formula to determine where it will place incremental spaces, but they don't guarantee that columns of data will remain aligned, since any number of incremental spaces may be placed within a column of data.

? AppleWorks Crash

Q I find that AppleWorks sometimes crashes when I delete characters from my word-processor documents. Is Apple aware of this?

A *In older versions of AppleWorks, attempting to delete a Page Break may crash the program if that command is the last item in the document. AppleWorks version 1.3 fixes this bug.* ■



Can We Afford a New House?

by John Slack

"Gee, I wish we had a new house. I want my own bathroom and closet, and there's no place to put that new computer!" With interest rates and housing costs dropping, the idea of owning your own home is very appealing. For current homeowners, it may mean the opportunity to move up to a larger house or to refinance the existing mortgage. In each instance, the most important factor is the amount of the new mortgage payment.

With the AppleWorks spreadsheet template presented here, you can determine your monthly mortgage payment, the amount of interest and principal paid, and the remaining balance on your loan. You can use the powerful AppleWorks-spreadsheet what-if capability to compare mortgage packages from different lending institutions and determine the best course of action.

The template produces a complete mortgage-repayment schedule. You enter amount of money borrowed, annual interest rate, term of the loan in years and months, and number of payments per year. Your Apple does the rest.

►Getting Started

Create a new spreadsheet file from scratch and call it MORTGAGE.SCHED. A blank spreadsheet containing rows and columns appears. Before entering titles, you must change column widths. Press OA-V (value), C (column width), then OA-right arrow three times, and Return.

Now position the cursor on cell A3. Type Mortgage Schedule and press Return. Skip two rows and go to cell A6. Type the following titles into cells A6 through A16: Principal, Term in Years and Months, Annual Interest Rate %, Number of Payments/Year, Decimal Year (calc), Tot Num of Paymnts (calc), Regular Payment (calc), Total Money Paid (calc), and Tot Interest Paid (calc).

Place your cursor on cell C6. Type " = and press Return. With the cursor still on cell C6, center this label: Press OA-L, E (entry), L (label format), and C (center). Now copy the label down column C to cell C16: Press OA-C, Return (staying within the worksheet), and Return again to indicate the source of the material to be copied. Move the cursor down one cell and type a period to indicate the start of the destination range. Move the cursor down again until it rests on cell C16. Press Return. Voilà—the labels appear. Blank out the extra label in cell C11: Press OA-B and Return.

To enter the titles for the mortgage schedule, go to cell A18 and type Payment; in cell A19 type Number. In the same manner, in cells B18/B19 type Interest Paid; in cells C18/C19 type Amount Amortized; in cells D18/D19 type Remaining Balance; in cells E18/E19 type Accumulated Interest; and finally in cells F18/F19 type Accumulated Equity.

To right-justify these labels, position the cursor on cell A18. Press OA-L (layout), B (block), down arrow once and right arrow five times to highlight the block, Return, L (label format), and R (right-justify). Then put the cursor on cell A20 and press "---- across the screen to underline the titles. At this time your screen should look like **Figure 1**.

►Entering Formulas

To calculate the amount of your regular mortgage payment, we use the following equation:

$$R = (I*P/N)/(1-((I/N)+1)^{-N*Y})$$

R = amount of regular payment; P = amount of mortgage; I = annual interest rate; Y = term in years; and N = number of payments per year.

The spreadsheet formula that represents this equation is too long to fit into one cell. To avoid this limitation, you can simplify the calculation process by breaking it into parts. In cell D12 calculate decimal years; in cell D13 calculate total number of payments. Position the cursor on cell D12, and enter the following formula:

(12*D7 + D8)/12

Press Return. Now place the cursor on cell D13, make this entry, and press Return:

+D10*((12*D7) + D8/12)

To enter the regular-payment formula, position the cursor on cell D14, type in the following, and press Return:

((.01*D9*D6/D10)/
(1 – (1/((.01*D9/D10)
+ 1)^(D12*D10))))

Now enter the following formulas in cells D15 and D16 and across line 21 (remember to press Return after each):

cell D15 +D13*D14
cell D16 +D15 – D6
cell A21 1 + A20
cell B21 .01*D6*D9/D10
cell C21 +D14 – B21
cell D21 +D6 – C21
cell E21 +B21 + E20
cell F21 +C21 + F20

Enter these formulas exactly as written—don't leave out any terms and don't omit any parentheses. When you're finished, your screen will show ERROR in cells D14 through D16 and across line 21. Don't worry—most formulas of this type display an error message when they have no data to process.

Now copy the formulas on line 21 to line 22: Put the cursor on cell A21 and press OA-C (copy), Return, right arrow five times to indicate the source, then Return. Now press the down arrow once to indicate the destination and press Return. The prompt goes through each term in the formulas in cells A22 through F22 and asks NO CHANGE or RELATIVE. Answer NO CHANGE for any formula item with a line number less than 19 (D12 and so on), and RELATIVE for all others.

Figure 1. *Entering labels.*

```
File: MORTGAGE.SCHED          REVIEW/ADD/CHANGE          Escape: Main Menu
=========A==========B==========C==========D==========E==========F======
  2¦
  3¦Mortgage Schedule
  4¦
  5¦
  6¦Principal                    =
  7¦Term in Years                =
  8¦   and Months                =
  9¦Annual Interest Rate %       =
 10¦Number of Payments/Year      =
 11¦
 12¦Decimal Year (calc)          =
 13¦Tot Num of Paymnts (calc)    =
 14¦Regular Payment (calc)       =
 15¦Total Money Paid (calc)      =
 16¦Tot Interest Paid (calc)     =
 17¦
 18¦     Payment      Interest      Amount   Remaining Accumulated Accumulated
 19¦      Number          Paid   Amortized     Balance    Interest      Equity
-------------------------------------------------------------------------------
A3: (Label) Mortgage Sched

Type entry or use @ commands                                  @-? for Help
```

Finally, change the formula in cell B22 to.01*D21*D9/D10; change D22 to + D21 – C22.

►Formatting Cells

The values you've entered on the template and the results AppleWorks calculates have two different formats. Term in Years and Months, Number of Payments/Year, and Total Number of Payments are integers with zero decimal places. All other items have two decimal places. The easiest way to format globally is to press OA-V, V, F, and 2 from anywhere in the spreadsheet. For individual cells D7, D8, D10, and D14, press OA-L, E, V, and F, then type a zero. Now reset the calculation order from columns to rows: Press OA-V, R, O (the letter), then R.

►Entering Data

After too many years living in the cramped quarters of Bachelor Manor, you decide to take the plunge and buy a house in Yuppie Glen. The real-estate salesperson says it's a steal at $125,000 and she can get you a 30-year mortgage at 9.25 percent. You have the $40,000 Aunt Willie left you, so you need an $85,000 mortgage. How much will your mortgage payments be, and what portion is tax-deductible interest?

To answer these questions, put the cursor on cell D6 and enter 85000. In cell D7 enter 30; in cell D8 type a zero; and in cell D9 enter 9.25. In cell D10 enter 12 to represent monthly payments. Your Apple tells you immediately that your regular monthly payment will be $699.27; total money paid will be $251,738.68, of which $166,738.68 will be interest if you live there the entire 30 years—not likely, since you're headed for Ritzy Oaks when you make Vice President.

To determine the amount of interest within your monthly payment, you must create a mortgage schedule. Copy line 22 as far into the future you care to go: Place the cursor on cell A22 and press OA-C (copy) and Return to stay within the spreadsheet. Press the right arrow five times to highlight the cells you want to copy, then press Return.

Figure 2. *Creating a mortgage schedule.*

```
File: MORTGAGE.SCHED          REVIEW/ADD/CHANGE          Escape: Main Menu
=========A==========B==========C==========D==========E==========F======
  4¦
  5¦
  6¦Principal                      =          85000.00
  7¦Term in Years                  =                30
  8¦   and Months                  =                 0
  9¦Annual Interest Rate %         =              9.25
 10¦Number of Payments/Year        =                12
 11¦
 12¦Decimal Year (calc)            =             30.00
 13¦Tot Num of Paymnts (calc)      =               360
 14¦Regular Payment (calc)         =            699.27
 15¦Total Money Paid (calc)        =         251738.68
 16¦Tot Interest Paid (calc)       =         166738.68
 17¦
 18¦     Payment      Interest       Amount   Remaining Accumulated Accumulated
 19¦      Number          Paid   Amortized     Balance    Interest       Equity
 20¦-------------------------------------------------------------------------
 21¦           1        655.21        44.07    84955.93      655.21        44.07
-------------------------------------------------------------------------
A4

Type entry or use @ commands                              @-? for Help
```

Figure 3. *Comparing mortgage alternatives.*

```
File: MORTGAGE.SCHED          REVIEW/ADD/CHANGE          Escape: Main Menu
=========A==========B==========C==========D==========E==========F======
  4¦
  5¦
  6¦Principal                      =          85000.00    85000.00
  7¦Term in Years                  =                30          35
  8¦   and Months                  =                 0           0
  9¦Annual Interest Rate %         =              9.25        9.15
 10¦Number of Payments/Year        =                12          12
 11¦
 12¦Decimal Year (calc)            =             30.00       35.00
 13¦Tot Num of Paymnts (calc       =            360.00      420.00
 14¦Regular Payment (calc)         =            699.27      675.95
 15¦Total Money Paid (calc)        =         251738.68   283896.95
 16¦Tot Interest Paid (calc)       =         166738.68   198896.95
 17¦
 18¦     Payment      Interest       Amount   Remaining Accumulated Accumulated
 19¦      Number          Paid   Amortized     Balance    Interest       Equity
 20¦-------------------------------------------------------------------------
 21¦        1.00        655.21        44.07    84955.93      655.21        44.07
-------------------------------------------------------------------------
A4

Type entry or use @ commands                              @-? for Help
```

Move the cursor down one line and type a period to indicate the start of the destination range.

Move the cursor down to line 380 if you want the schedule for the entire 30 years, and press Return. The prompt goes through all terms of the formulas and asks NO CHANGE or RELATIVE. Answer NO CHANGE for any item with a line number less than 20 and RELATIVE for the rest. Your Apple takes several moments to set up the spreadsheet and perform the calculations. When it's finished, your screen will look like **Figure 2**. Note that a spreadsheet this large takes up a lot of RAM—be sure you have enough desktop space.

►Playing "What If?"

One of your electronic spreadsheet's most powerful features is its ability to immediately provide the new result of a calculation whenever you change a parameter, letting you compare results very quickly and easily.

Let's compare the results of different mortgage alternatives. Place the cursor on cell D6. Press OA-C, then Return to keep the copy within the worksheet. At the prompt to indicate the source, move the cursor down to cell D16 and press Return. The prompt now asks the destination. Place the cursor on cells E6, F6, G6, and so on, and press Return. The prompt goes through each term in the formulas in cells D12 to D16 and asks NO CHANGE or RELATIVE. Answer RELATIVE for each term. The screen now shows the data in column D repeated in columns E, F, G, and so on. To evaluate other mortgage offerings, just change the data shown.

Since you're a bright, up-and-coming executive, you find several banks besides Tightfisted Savings and Loan (recommended by your real-estate agent) that are willing to loan you the money for your new house. Wegotcha Mortgage Finance offers 9.15 percent on 35-year loans. To see how it compares, place the cursor on cell E7 and change 30 to 35. In cell E9 change 9.25 to 9.15. Your screen now says your mortgage payment would be $675.95. The totals are worse, though—if you stay 35 years. At this time your screen should look like **Figure 3**.

To change the mortgage schedule, blank out cells D6 through D16 and copy cells E6 through E16 in their place. All formula items are RELATIVE. Save the template to a data disk and recall it each year when you make out your tax return: Mortgage interest is deductible, and in the early years of a mortgage most of the money goes toward interest. Now you know why owning your own home is such a good tax shelter!■

How Much Time?

Continued from p. 1

The way the AppleWorks database handles times and dates is one of its slickest features. AppleWorks automatically converts anything you enter into a database-report category with Date or Time in its name to its time/date format (MAR 15 87, 12:37 PM). AppleWorks likes to have times and dates in a standard

format so that it can sort time and date categories chronologically—that, luckily, is AppleWorks' sole built-in time/date calculation. Like a well-trained horse, the program tries to obey you even when you don't issue the commands correctly. Type almost anything into a "date" or "time" category and AppleWorks will probably know what you intended, regardless of what you typed. For example, hours and minutes are supposed to be separated by colons, but you can use semicolons if you don't want to hold down the shift key.

Like all labor-saving devices, though, this one has drawbacks as well as advantages.

AppleWorks can't automatically subtract one date from another to give you the answer. That's because AppleWorks-standard dates and times are stored as standard ASCII text characters. This makes it impossible to perform even simple arithmetic.

If you enter numeric data into categories that don't contain the words "time" or "date," you can perform simple arithmetic, but you can't sort chronologically.

The following two macro techniques show you how to combine the chronological sorting strength of the AppleWorks database with the number-crunching power of the AppleWorks spreadsheet. But, if you create database files from DIF or ASCII files, not everything you put into a category called Date or Time can be sorted chronologically. AppleWorks chronologically sorts only database categories it created.

Try printing an AppleWorks database with a time category to disk, then creating a new one from the ASCII file on disk. After you rename all categories, you'll have something that *looks* exactly the same as the original, but isn't. The "chronological" and "reverse chronological" choices will appear whenever you want to sort the time categories, but they won't work.

This time and date blindness occurs whenever you create a database from an ASCII or DIF file, so when you import spreadsheet times or dates into the database you can't automatically take advantage of the chronological-sort feature.

You can fool AppleWorks into sorting times and dates properly by pretending that you've typed in all the entries yourself. You'll have to retype the entry, or at least one character, to get AppleWorks to wake up. A macro that inserts, then deletes, a leading or trailing space will accomplish the job, although AppleWorks executes this so quickly you may have to sort the records to assure yourself it really happened.

Here's a macro that adds and promptly deletes a space in the time category. Use it on Time in multiple-record layout:

```
<right><right><right>
<right><right><right>
<right><right><space>
<delete><return><macro>
```

The macro spaces to the end of the time category, adds a space, deletes it, moves to the next record, and repeats itself. For dates, use a similar macro on the date category, but use nine right arrows instead of seven:

```
<right><right><right>
<right><right><right>
<right><right><right>
<space><delete><return>
```

This solves the simple problem of ensuring that database dates and times are indeed in Date/Time format. But converting database dates and times to numeric form so the spreadsheet can use them to compute elapsed hours or days is considerably more complicated. Try doing this for even a small database by hand, and you'll see the wisdom of getting a macro program and a RAM disk.

And even after you put dates and times into the spreadsheet, you'll remember why you hated eighth-grade math so much: Doing arithmetic with nondecimal numbers, such as the base-12 time system or the baseless calendar, is no day at the beach.

Most of your problems have to do with borrowing and carrying, so if you can minimize the amount required, the solution will be simpler. Generally speaking, the longer the time between two dates, the more complicated the arithmetic.

And it may be better to be wrong by a day once every four years than to spend eight hours on a formula to handle leap years.

►Elapsed Time

I'll explain in detail how to compute the time elapsed between two events that take place during a single business day—a technique useful to lawyers, consultants, auto machanics—anyone who bills by the hour. Figuring the number of days passed is a similar job, but harder.

If you start a job at 4:30 in the afternoon and don't finish until after the bars close, this solution won't work because it doesn't allow for borrowing 24 hours to make the subtraction work.

The first step is to prepare a report format that contains

only the "time begun" category, and transfer the data to the word processor by printing the report to the clipboard, then copying it to a temporary word-processing file. Delete all the information at the top: name of report, date of report, and so on. You should see times like these in your word-processing file:

4:46 PM
8:30 AM
2:00 PM
9:00 AM
4:50 PM

You want to change them to this format:

4
:
46
PM
8
:
30
AM

And so on. This change ensures that each arithmetic element in the date or time will become a separate category or column for the database or spreadsheet. Don't worry about the colons.

Here's how to do it with a word-processing macro:

<right><right><return>
<right><return><right>
<right><return> <down>
<macro name>

Next, use AppleWorks' search-and-replace feature to convert all instances of AM to 0 (zero) and all PMs to 12. Later, in the spreadsheet, you'll add this value to the time to produce 24-hour times. With dates, similarly use search-and-replace to convert all the months to numeric equivalents.

Before you move the data to the spreadsheet, you have to move from the word processor to the database, by printing the word-processing file as an ASCII text file and making a new database file from that ASCII file. (Now you know why a RAM disk is such a help in this technique.)

When you create the new database file, answer the question "how many categories?" with 4. You'll have a colon category, but don't worry. Because the information was evenly spaced when you printed it to disk, it will flow evenly into categories for "hours," "colons," "minutes," and "zero or twelve," plus an empty one.

Don't take the time, though, to rename your categories, because you aren't going to stay in the database long enough to get to know them.

Create a report format that includes the hours, minutes, and "zero or twelve" categories (probably one, three, and four), and print it as a DIF file on disk.

Then, after you create a new spreadsheet from the DIF file, repeat all the above steps with the "time ended" data from your original database. Finally, copy the data from one of the two spreadsheets to the other via the clipboard.

You'll wind up with two sets of data in rows that look like the accompanying **Figure 4**.

The next step is to convert hours to 24-hour time. Then multiply by 60 to convert hours to minutes, and add the result to the minutes to get the total number of minutes. Put the formula ((A5 + C5)*60) + B5 into D5, and copy it (using the relative copy) to D6, D7, and so on.

Next construct a formula that subtracts your first starting time from your first ending time. One that gives you a decimal result is (D13 – D5)/60.

When you're sure everything's working, copy the formula into the cells representing your second through *n*th starting times to produce something like the accompanying **Table 1**.

You can use the same basic process to subtract days and months, although it's a little more complicated. Use **Table 2** to convert months to days.

By adding the value in the days column to the day of the month, you can determine

Figure 4. *Time data in the AppleWorks spreadsheet.*

	A	B	C	D	E
1			Time Begun		
2					
3	Hours	Minutes	(0/12)		
4					
5	4	46	12		
6	8	30	0		
7	2	0	12		
8	9	0	0		
9	4	50	12		
10					
11			Time Ended		
12					
13	7	23	12		
14	4	22	12		
15	2	18	12		
16	1	10	12		
17	6	12	12		
18					

Table 1. *Figuring elapsed time.*

Time Begun				
Hours	Minutes	(0/12)	Tot. Min.	Elapsed Time
4	46	12	1006	2.62
8	30	0	510	7.87
2	0	12	840	.30
9	0	0	540	4.17
4	50	12	1010	1.37
			0	
	Time Ended		0	
			0	
7	23	12	1163	
4	22	12	982	
2	18	12	858	
1	10	12	790	
6	12	12	1092	

Table 2. *Converting months to days.*

Month		Days
Jan	1	0
Feb	2	31
Mar	3	59
Apr	4	90
May	5	120
Jun	6	151
Jul	7	181
Aug	8	212
Sep	9	243
Oct	10	273
Nov	11	304
Dec	12	334

the number of days from the first of the year a given date represents. By subtracting any two such numbers, you can determine the number of days between any two dates within the year.

The current wave of AppleWorks macro programs changes these techniques from an interesting theoretical possibility into a practical reality. Using a RAM disk and macros, I transferred and converted the five records in my example in only 22 seconds.

And, by the way, if your boss left for lunch at 11:47 a.m. and returned at 2:18 p.m., he was gone 2.52 hours. Rank indeed hath its privileges. ■

New Products

Program Selectors

by Paul Statt

Many AppleWorks users now have high-capacity storage devices. Three-and-a-half-inch disk drives with 800K disks and 5-, 10- or 20-megabyte hard disks hold more information than your tired old floppies, but "more" doesn't always mean "easy to find."

Program selectors make it simple to locate programs and files on your high-capacity drives. The three described below are powerful and inexpensive (one of them is free). They work with any type of storage device, but shine when you have more storage than two floppy drives.

ProSEL is my favorite. Glen Bredon's nifty handiwork installs simply and works smoothly and efficiently. ProSEL becomes the first .SYSTEM program on your hard disk, and boots fast because it's small. You see a menu that's easy to use, even without AppleWorks format or mouse support: Just scroll through your choices.

One option is a ProSEL menu editor, with which you can add items to the selection. You'll probably add every program on your hard disk—it's as easy as typing each pathname once, then giving your selection another name. Go back to the menu: There, your program's now an option.

You enter a program by highlighting its name and pressing Return. When you quit, you go back to the menu. If you're like me, calling up four versions of AppleWorks, Point to Point, and CommWorks during the course of a day, ProSEL is a must.

The ProSEL disk has something extra for everyone—a volume-copy program that's faster than the Filer, a ProDOS disk-block editor, and a tree-structure catalog that's great if you're knee-deep in hard-disk subdirectories and sinking fast.

ProSEL is a bargain—$40, mail-order only from Glen Bredon, 521 State Road, Princeton, NJ 08540. Please don't try to call—trust us, he'll send ProSEL or return your money.

ProSEL's inexpensive, but **Bird's Better BYE** is free. Whenever you buy any program from Bird's current company (The Software Touch, 9842 Hilbert Street, San Diego, CA 92131, 619-549-3091) or his former employer (Beagle Bros, 3990 Old Town Avenue, San Diego, CA 92110, 619-296-6400), you'll find that the PRODOS file on the disk is slightly altered. It's the BYE command that's changed.

The ProDOS BYE command is the one that orders you to "Enter prefix" and "Press carriage return to accept," with all the charm of a Parris Island drill sergeant, when

you quit AppleWorks. Bird's BYE is friendlier. It tells you the current ProDOS prefix and lists all the .SYSTEM files on your volume. Since that's probably not the volume you want (you just quit, after all), a quick "escape" changes the prefix and lists the .SYSTEM files on that volume.

You can boot by highlighting a filename and pressing Return; Bird's Better BYE even delves into subdirectories. With a touch of BASIC programming (you could buy **Program Writer**—$39.95, by Alan Bird—from The Software Touch and learn), you can turn Bird's Better BYE into Bird's Best HELLO, a program that lists names of available programs when you start up in the morning.

You can't beat the price, and Bird's Better BYE works especially well if you have a number of disk drives of varying capacities—say, a DuoDisk and one 3½-inch UniDisk.

The dark horse among program selectors is **Nite Owl Productions Director**, one of seven programs on Nite Owl's Developer Disk #2 ($39.95, Nite Owl Productions, 5734 Lamar Avenue, Mission, KS 66202, 913-362-9898).

The Director adds a number of commands to the ProDOS BASIC.SYSTEM language—author Bob Shofstall's specialty—among them, DIR. DIR displays a tree directory of all ProDOS storage devices on line and lets you set the system prefix. A tree directory, for the uninitiated, is the kind you see in "Introduction to ProDOS" articles, and never again. ProSEL has one, too. It makes sense of ProDOS and looks like this:

```
/correspondence
  /business.letters
    /incider
    /main.menu
  /personal.letters
    /mom
    /dad
  /misc.letters
```

Bird's Better BYE doesn't let you see the whole tree at once, but at least you can crawl up all the branches. That's the power of ProDOS program selectors: They let you see what's available anywhere in your system. They're great buys—every AppleWorks user should have one. ■

Focus on Function

Formatting Empty Spreadsheet Cells

by Kelly Stirn

Have you ever used AppleWorks' layout commands to format a row or column of figures in a spreadsheet, only to find out later when you tried to use it that in blank cells AppleWorks set no format at all? Here's a simple explanation, and a trick to help you get around the problem.

Let's say you're creating a financial report, with one column of data you'd like to format for "Dollars" and another for "Percentage." To format the Dollars column, you'd probably type the layout command, OA-L, then select Columns, Value Format, and Dollars. What you'd find, though, is that OA-L formats only cells containing data; cells that were blank when you set the layout remain unformatted.

Here's the solution: Select the Block command, which formats all cells, including blank ones, instead of Columns or Rows, then highlight all cells you predict will be filled with data. You can always highlight additional ones later.

It's easy to select an entire row or column when you actually want to format only a small portion of it. Giving every cell a particular format attribute uses up valuable memory, but AppleWorks luckily protects you from wasting your RAM. ■

Favorites

►Ditto, Ditto

A tip for AppleWorks database users: The ditto command (Open apple-') works when you move the cursor *across* columns as well as down. ■

—Eugene M. Hora, San Diego, California

Reviews

Super Macros

by Paul Statt

This one took my breath away. I booted AppleWorks and saw not the main menu, but a brand new word-processor file (called "New") and this charming note:

"Super MacroWorks Copyright (c) 1986, Randy Brandt
Beagle Bros, Inc.
It's February 19, 1987, and the time is 20:42. Press Return to get started."

I pressed Return, and I started this review. This, I recalled dimly from my innocent youth, is what computers are supposed to do: AppleWorks is actually doing my work for me—creating a new file and getting me started. And it actually knows something about the "real world"—the time and date.

That's all I ever wanted a computer or AppleWorks to do, and **Super MacroWorks** does both. It offers something to AppleWorks newcomers as well as jaded old-timers: macros that make life easier for beginners and a proto-programming language for pros.

►What's a Macro?

A *macro* is a single keystroke that replaces many. Imagine AppleWorks as a puppy that knows "Pick up," "Come," and "Drop." Unless you want to yell three things every time you want your slippers, you'll have to teach him a "macro-command" that replaces "Pick up," "Come," and "Drop" with the easier-to-say "Slippers." That's a macro.

A macro can't teach an old dog new tricks: You can string together only legitimate AppleWorks commands into a macro. But with Super MacroWorks you create solid-apple commands that complement the open-apple commands that come with AppleWorks.

The most obvious use of a macro is to save you the work of typing the same thing over and over again. Writing this review, for example, I defined a solid-apple command to spell "Super MacroWorks." That might be reason enough for some AppleWorks people to spend $34.95, but I could have gotten the same result by typing a slash or some other unusual character wherever I wanted "Super MacroWorks" and putting AppleWorks' find-and-replace command (Open apple-R) to work on the document when I finished. What's the advantage of a separate macro program?

A macro can do the work of *any* AppleWorks keystroke—it doesn't just type. For instance, it can get the first file from your current data disk. You define Solid apple-G to "<escape><return><return><return>." When you press Solid apple-G, AppleWorks does just what you told it to: escapes to the main menu and selects the "default" setting on the next three menus (Add Files to the Desktop, Get Files from the Current Disk, and the first file on that floppy).

But any macro program can replace any AppleWorks key sequence with a single stroke. And all macro programs now offer two kinds of macros. Temporary macros ("recorded" macros in Super MacroWorks) are ones you create but don't save. Permanent macros, usually called "compiled," as in Super MacroWorks, are those you save on your AppleWorks disk. You must compile them into assembly language, the language in which AppleWorks is written.

►Super Performance

So what's the advantage of Super MacroWorks? First, it's perfectly attuned to AppleWorks 2.0. Updated versions of other macro programs also work with 2.0, but Super MacroWorks and all its functions were designed just for 2.0.

But the real advantages of any macro program are its extras, and Super MacroWorks has lots. You can program a startup macro, for instance (such as the greeting above). If you're aching to use a mouse in AppleWorks, Super MacroWorks lets you. It's the first AppleWorks mouse I've seen that lets you scroll through word-processor documents by holding down the mouse button.

You get Bird's Better BYE, a ProDOS patch that serves as a simple file manager; you can rewrite the AppleWorks help screens; you can change the AppleWorks error buzz to a beep. Alan Bird's patch that eliminates the wait for disk swapping at bootup, if you have a hard-disk drive or 3½-inch drive, is also included. The most important extras are the ones that allow for macros that can do more than a series of AppleWorks commands can.

For instance, no AppleWorks command can add one to the number representing the cur-

sor's position, print the time and date from your ProDOS clock into a file, change uppercase to lowercase in a file, or print only the line on which the cursor rests just as easily as Open apple-H prints a hardcopy of your screen.

Super MacroWorks' new AppleWorks commands can do all that and a lot more. Super MacroWorks adds logic to AppleWorks macros with if-then statements that can make decisions in a series of AppleWorks commands: "If the character in a spreadsheet column is 1, then highlight that character and copy it to the clipboard," for instance.

►Checking Out the Competition

Super MacroWorks doesn't work if you use Pinpoint desktop accessories: With Pinpoint you'll need Pinpoint's own KeyPlayer or The Software Touch's AutoWorks. You can make any macro program, including Super MacroWorks, work with whatever memory-expansion card you have—Apple's, Applied Engineering's RamWorks, Checkmate's MultiRam, or any other. Compared to Super MacroWorks, though, KeyPlayer's installation is fussy: You have to install it on an unmodified Pinpoint, which in turn you must install on an unmodified AppleWorks disk, unless you have RamWorks memory expansion, which you have to install first.

Super MacroWorks is a rudimentary programming language. It's simple, but that doesn't excuse Beagle Bros' lack of explanation in an otherwise-excellent manual. Beagle Bros' books are always witty and packed with information; the Super MacroWorks documentation is, too. But this time the product is likely to be in the hands of a nonprogrammer, who might need a little more help. (Pinpoint's KeyPlayer also adds logical macros, but that manual is more confused than Beagle Bros'.)

Randy Brandt, author of MacroWorks, is a super programmer. All the changes to AppleWorks that make Super MacroWorks special require assembly language in the AppleWorks code. The sad truth about Super MacroWorks—and Super MacroWorks is reason enough to buy AppleWorks 2.0—is that Apple could have put everything into AppleWorks 2.0 that Super MacroWorks does. Apple didn't, but Beagle Bros did. For $34.95 you can turn AppleWorks 2.0 into a new AppleWorks.■

Product Information

Super MacroWorks
Beagle Bros
3990 Old Town Avenue
San Diego, CA 92110
(619) 296-6400
$34.95
Rating: 🍎🍎🍎🍎🍎

Favorites

►What's the DIF?

I like the easy entry of an AppleWorks-spreadsheet checkbook template ("Where Does All the Money Go?", February 1987, p. 4) and the easy reporting of a database checkbook register. You can combine both with a data-interchange-format (DIF) file.

You can print the number, date, recipient, and amount of the check from the spreadsheet to a DIF file on disk (including your RAM disk). (Make sure to print by columns, not rows.) Then create a database from that DIF file. It's almost as quick as using the clipboard with a RAM disk.

In the database, reports are much more flexible—you can sort checks by date or number, and use both to balance your monthly statement. If you include other spreadsheet columns, such as a coded zero or one for business or personal expenses, it's simple to sort on that category in the database version.

A DIF file keeps your columns and rows straight and turns them into neat database categories. Use it often to keep track of your expenses.■ **—Paul Statt, Contributing Editor**

The Main Menu reviews range from poor to excellent denoted by one to five red apples. If a product receives one red apple, it means the reviewer wouldn't spend his money to buy it and neither should you. If it receives five red apples, it means that our usually tight-fisted reviewer would part with his money to buy it because of the value it offers in increased productivity and performance.

excellent 🍎🍎🍎🍎🍎
good 🍎🍎🍎🍎
satisfactory 🍎🍎🍎
fair 🍎🍎
poor 🍎